

REMARKS

This application has been carefully reviewed in light of the Office Action dated April 7, 2006. As indicated above, claims 15 to 20 have been cancelled herein, without prejudice or disclaimer of subject matter. Claims 1 to 14 remain in the application, of which claims 1, 3 and 9 to 14 have been amended herein. Claims 1, 9 and 14 are the independent claims. Reconsideration and further examination are respectfully requested.

In the Office Action, claim 15 was objected to for an alleged informality. Since claim 15 has been cancelled herein, without prejudice or disclaimer of subject matter, the applicant submits that the objection is rendered moot.

Claims 1 to 20 were rejected under 35 U.S.C. § 102(b) over M. Carey et al, "On Saying 'Enough Already!' In SQL," PROCEEDINGS OF THE 1997 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA (Tucson, AZ)(Vol. 26, No. 2, Jun. 1997, pp. 219-30) ("Carey"). As indicated above, claims 15 to 20 have been cancelled herein, without prejudice or disclaimer of subject matter, and without conceding the correctness of the rejection. Furthermore, independent claims 1, 9 and 14 have been amended herein to further clarify the features that *i*) the sorted result buffer is iteratively ordered based upon the order criteria, *ii*) remaining records in the data store are iteratively compared against a *N*th record in the sorted result buffer based upon the order criteria, and *iii*) the *N*th record in the sorted result buffer is iteratively replaced with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the *N*th record in the sorted result buffer. Support for these newly clarified features is described throughout the disclosure, including at least pages 7 to 10 of the specification, and Figure 5. Withdrawal of the rejection and further examination are therefore respectfully requested.

The present disclosure generally relates to the satisfaction of limit and order queries. A limit and order query that includes both of an order criteria and a limit criteria are received, the limit criteria specifying a maximum number *N* of records for a result set of records satisfying the limit and order query. A sorted result buffer is filled with a first *N* number of records from a data store, and the sorted result buffer is iteratively ordered based upon the order criteria. Remaining

records in the data store are iteratively compared against a N th record in the sorted result buffer based upon the order criteria, and the N th record in the sorted result buffer is iteratively replaced with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the N th record in the sorted result buffer. The sorted result buffer is output as the result set of records.

Referring to particular claim language, independent claim 1 recites a data store query system. The system includes a data store that includes a collection of records, a sorted result buffer, and a query interface. The query interface is operable to receive a limit and order query that includes both of an order criteria and a limit criteria, the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query. The query interface is also operable to fill the sorted result buffer with a first N number of records from the data store, iteratively order the sorted result buffer based upon the order criteria, to iteratively compare remaining records in the data store against a N th record in the sorted result buffer based upon the order criteria, and to iteratively replace the N th record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the N th record in the sorted result buffer. The query interface is further operable to output the sorted result buffer as the result set of records.

Independent claim 9 recites a method for satisfying limit and order queries. The method includes receiving a limit and order query that includes both of an order criteria and a limit criteria, the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query, and filling a sorted result buffer with a first N number of records from a data store. The method also includes iteratively ordering the sorted result buffer based upon the order criteria, and iteratively comparing remaining records in the data store against a N th record in the sorted result buffer based upon the order criteria. Furthermore, the method includes iteratively replacing the N th record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the N th record in the sorted result buffer, and outputting the sorted result buffer as the result set of records.

Independent claim 14 recites an apparatus including a storage medium having instructions stored thereon. The instructions include a first code segment for receiving a limit

and order query that includes both of an order criteria and a limit criteria, the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query, a second code segment for filling a sorted result buffer with a first N number of the set of data records from a data store, and a third code segment for iteratively ordering the sorted result buffer based upon the order criteria. The instructions also include a fourth code segment for iteratively comparing remaining records in the data store against a N th record in the sorted result buffer based upon the order criteria, a fifth code segment for iteratively replacing the N th record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the N th record in the sorted result buffer, and a sixth code segment for outputting the sorted result buffer as the result set of records.

The applied art is not seen to disclose, teach, or to suggest the foregoing features recited by the independent claims. In particular, Carey is not seen to disclose at least the features that *i*) the sorted result buffer is iteratively ordered based upon the order criteria, *ii*) remaining records in the data store are iteratively compared against a N th record in the sorted result buffer based upon the order criteria, and *iii*) the N th record in the sorted result buffer is iteratively replaced with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the N th record in the sorted result buffer.

Carey describes a proposed SQL extension that would allow a query writer to limit the cardinality of a query result, such as a query which processes at most some number of result tuples. *See Carey, Abstract and Introduction.* Using the proposed STOP AFTER language, for example, only a specified number of result tuples would be retained. *See Carey, Extending SQL.* While it is true that Carey, as well as numerous other previously-disclosed references, recognize that it is beneficial to limit the number of results in a query, it is also true that the particular approaches described by Carey differ greatly than those described by the present invention. In particular, in one arrangement of Carey, a specified number of tuples are inserted into a priority heap, and the remaining tuples are tested with the heap's membership to determine whether or not the new tuple's value warrants its insertion the heap. *See Carey, Sort-Stop.* This approach tests remaining tuples against each member of the priority heap, and is seen to be distinguishable from the present invention, in which each record in the data store is merely iteratively compared

against the N th record in the sorted result buffer, where it is possible to limit such testing to one record due to the iterative ordering of the sorted result buffer (based upon the order criteria).

Accordingly, Carey is not seen to disclose, nor does the Office Action even allege that Cary discloses, at least the features that *i*) the sorted result buffer is iteratively ordered based upon the order criteria, *ii*) remaining records in the data store are iteratively compared against a *N*th record in the sorted result buffer based upon the order criteria, and *iii*) the *N*th record in the sorted result buffer is iteratively replaced with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the *N*th record in the sorted result buffer.

Accordingly, based on the foregoing amendments and remarks, independent claims 1, 9 and 14 are believed to be allowable over the applied references. The remaining rejected claims are each dependent on these independent claims and are believed to be allowable for at least the same reasons. Because each dependent claim is deemed to define an additional aspect of the invention, individual consideration of each on its own merits is respectfully requested.

No other matters being raised, it is believed that the entire application is fully in condition for allowance and such action is courteously solicited.

No fees are believed to be due at this time. Please apply any other charges or credits to deposit account 06 1050.

Respectfully submitted,

Date: JUNE 16, 2006

David E. A. Jordan
Reg. No. 50,325

Fish & Richardson P.C.
1425 K Street, N.W.
11th Floor
Washington, DC 20005-3500
Telephone: (202) 783-5070
Facsimile: (202) 783-2331